Computing 22, 367—373 (1979)

Computing
© by Springer-Verlag 1979

Algorithm 39

Clusterwise Linear Regression

H. Späth, Oldenburg

Received August 3, 1978

Abstract — Zusammenfassung

Algorithm 39. Clusterwise Linear Regression. The combinatorial problem of clusterwise discrete linear approximation is defined as finding a given number of clusters of observations such that the overall sum of error sum of squares within those clusters becomes a minimum. The FORTRAN implementation of a heuristic solution method and a numerical example are given.

Algorithmus 39. Klassenweise lineare Regression. Die kombinatorische Aufgabe der klassenweisen diskreten linearen Approximation wird dadurch definiert, daß die Summe über die Fehlerquadratsummen innerhalb der Klassen minimiert wird. Die FORTRAN-Implementation eines heuristischen Lösungsverfahrens und ein numerisches Beispiel werden angegeben.

1. Problem

The usual form of linear least squares regression can be described as follows. Let be given m observations (y_i, a_{ik}) $(i=1, \ldots, m, k=1, \ldots, l)$ with $m>l$. Determine $(x_1, \ldots, x_l)$ such that

$$\sum_{i=1}^{m} \left(y_i - \sum_{k=1}^{l} a_{ik} x_k \right)^2 \tag{1}$$

is minimized.

But if the observations ought to be assigned to different unknown groups due to not collected, not collectable or unknown values for further independent variables, then, especially in the case of $m \gg l$, the following problem formulation seems to be more adequate:

Find a partition $C_1, \ldots, C_n$ of given length n of the observations, i.e.

$$C_j \subset M = \{1, \ldots, m\}, \ |C_j| > 0, \ C_j \cap C_k = \emptyset \text{ for } j \neq k, \ \bigcup_{j=1}^{n} C_j = M,$$

and n vectors $(x_{j1}, \ldots, x_{jl})$ $(j=1, \ldots, n)$ such that

$$D(C_1, \ldots, C_n) = \sum_{j=1}^{n} E(C_j) \tag{2}$$

0010-485X/79/0022/0367/$ 01.40

is minimized, where
$$E(C_j)=\sum_{i\in C_j}\left(y_i-\sum_{k=1}^{l} a_{ik}x_{jk}\right)^2. \tag{3}$$

In order to ensure a necessary condition for the partial problems (3) to have a solution we have to add the side conditions

$$|C_j|\geq l \quad (j=1,\ldots,n). \tag{4}$$

This problem formulation is related to the minimum variance criterion from cluster analysis [14] and is more generally given in [16]. A corresponding practical problem is given in [6].

2. Method

As the number of possible partitions is too high to enumerate them within reasonable computing times for realistic sizes of m and l, one has to choose some heuristic solution method that gives a good but not necessary an optimal solution. As the so-called exchange method empirically behaves very well for similar structured cluster problems [14, 15], it looks attractive to modify it with regard to (4) for this problem:

Step 1: Choose some initial partition with property (4). Set $i:=i_0, i_0\in M$ (e.g. $i_0=m$).

Step 2: Set $i:=i+1$ and reset $i:=1$ if $i>m$. Let be $i\in C_j$. Then for $|C_j|>l$ it is examined if there are clusters C_p with $p\neq j$ such that

$$E(C_p\cup\{i\})+E(C_j-\{i\})<E(C_p)+E(C_j). \tag{5}$$

If this is true, then let r be an index such that the reduction of (3) corresponding to (5) becomes a maximum. In this case redefine

$$C_j:=C_j-\{i\},\quad C_r:=C_r\cup\{i\}.$$

In all other cases return to step 2.

Step 3: Repeat step 2 as long as the objective function (3) can be reduced, e.g. as long as i has been increased m times without (5) being true.

This method is stepwise optimal and works sequentially on the observations. Its result depends on the initial partition. As with cluster problems [14] it happens empirically that after 6 to 10 passes through the observations one obtains a useful solution. It is recommended to start with several initial partitions and to select the best solution.

3. Algorithm

The above method is implemented in the following subroutine CWDLR whose formal parameters are described precisely in the comment cards of the list. An initial partition $C_1,\ldots,C_n$ has to be given via an integer vector $(p_1,\ldots,p_m)$ with $p_i=j$ if observation i shall belong to cluster C_j. For solving the problems of type (3) the very reliable subroutine HFTI from [8] is used. If the rank of the matrix (a_{ik}) $(i\in C_j,\ k=1,\ldots,l)$ should be less than l, then HFTI gives a minimal length solution.

List

```
      SUBROUTINE CWDLR (A,MDA,M,L,Y,P,X,NDX,N,D,E,
     *                  Q,IP,AA,YY,YA,YB,F,G,IA)
C
C     THE SUBROUTINE PARAMETERS ARE DEFINED AS FOLLOWS:
C
C     A(MDA,L)  THE ARRAY A HAS TO CONTAIN THE GIVEN
C               (M,L)-MATRIX (M<=MDA)
C     Y(M)      THE ARRAY Y HAS TO CONTAIN THE GIVEN
C               M-VECTOR
C     P(M)      THIS M-VECTOR (OF TYPE INTEGER) INIATIALLY
C               HAS TO CONTAIN A FEASIBLE PARTITION OF
C               LENGTH N VIA P(I)=J (I=1,...,M).
C               ON OUTPUT P CONTAINS THE FINAL PARTITION
C               OBTAINED BY THE EXCHANGE METHOD
C     X(NDX,L)  WILL CONTAIN THE (N,L)-MATRIX (N<=NDX) OF
C               SOLUTION PARAMETERS, I.E. X(J,*) IS THE
C               FOUND PARAMETER SET FOR THE J-TH CLUSTER
C               (J=1,...,N)
C     D         THE OBTAINED VALUE OF THE OBJECTIVE FUNCTION
C     E(N)      E(J) WILL CONTAIN THE ERROR SUM OF SQUARES
C               FOR THE J-TH CLUSTER
C
C     THE OTHER ARRAYS ARE FOR WORKING SPACE. THEIR
C     LENGTHS CAN BE SEEN FROM THE DIMENSION STATEMENTS.
C
C     IA        THE EXCHANGE STEPS ARE STARTED AT THE ROW
C               NUMBER IA + 1 (MOD M). NORMALLY ONE SETS IA=0.
C
C     THE VERY RELIABLE SUBROUTINE 'HFTI' IS USED TO SOLVE ALL
C     LEAST SQUARES PROBLEMS. A LIST CAN BE FOUND IN:
C     LAWSON/HANSON, SOLVING LEAST SQUARES PROBLEMS,
C     PRENTICE HALL 1974.
C
      DIMENSION A(MDA,L), Y(M), X(NDX,L),E(N),
     *          AA(MDA,L),YY(M),YA(L),YB(L),F(M),G(M),RNORM(1)
      INTEGER   P(M),Q(N),IP(L),H,U,V,R
      LOGICAL ITRUE
      TAU=1.E-10
      ITRUE=.TRUE.
      I=IA
      IT=0
      D=0.
    1 DO 10 J=1,N
            IF(ITRUE) GOTO 2
            IF(J.NE.R) P(I)=J
            IF(J.EQ.R) P(I)=0
    2       V=0
            DO 4 H=1,M
                 IF(P(H).NE.J) GOTO 4
                 V=V+1
                 YY(V)=Y(H)
                 DO 3 K=1,L
                      AA(V,K)=A(H,K)
    3            CONTINUE
    4       CONTINUE
            IF(V.LT.L) RETURN

C
            CALL HFTI (AA,MDA,V,L,YY,1,1,TAU,KRANK,RNORM,F,G,IP)
C
            FF=RNORM(1)*RNORM(1)
            IF(.NOT.ITRUE) GOTO 6
            Q(J)=V
            E(J)=FF
            D=D+FF
            DO 5 K=1,L
                X(J,K)=YY(K)
    5       CONTINUE
            GOTO 10
    6       IF(J.NE.R) GOTO 8
            AF=FF
            DO 7 K=1,L
                 YA(K)=YY(K)
```

List (Continuation)

```
 7        CONTINUE
          GOTO 10
 8        IF(FF.GT.BF) GOTO 10
          BF=FF
          U=J
          DO 9 K=1,L
               YB(K)=YY(K)
 9        CONTINUE
10 CONTINUE
   IF(.NOT.ITRUE) GOTO 11
   IF(N.LE.1) RETURN
   ITRUE=.FALSE.
   GOTO 14
11 BU=BF-E(U)
   RA=E(R)-AF
   IF(BU.LT.RA) GOTO 12
   IT=IT+1
   P(I)=R
   GOTO 14
12 IT=0
   E(R)=AF
   E(U)=BF
   D=D-RA+BU
   P(I)=U
   Q(R)=Q(R)-1
   Q(U)=Q(U)+1
   DO 13 K=1,L
          X(R,K)=YA(K)
          X(U,K)=YB(K)
13 CONTINUE
14 I=I+1
   IF(I.GT.M) I=I-M
   IF(IT.EQ.M) RETURN
   R=P(I)
   IF(Q(R).LE.L) GOTO 14
   BF=1.E50
   GOTO 1
   END
```

4. Numerical Example

For the data from Table 1 it has been started with the standard initial partition $p_i = 1 + \mathrm{mod}\,(i-1, n)$ and $i_0 = m = 20$.

Table 1

i	y_i	a_{i1}	a_{i2}	a_{i3}
1	960.0	60.0	18.0	1.0
2	830.0	220.0	0.0	1.0
3	1260.0	180.0	14.0	1.0
4	610.0	80.0	6.0	1.0
5	590.0	120.0	1.0	1.0
6	900.0	100.0	9.0	1.0
7	820.0	170.0	6.0	1.0
8	880.0	110.0	12.0	1.0
9	860.0	160.0	7.0	1.0
10	760.0	230.0	2.0	1.0
11	1020.0	70.0	17.0	1.0
12	1080.0	120.0	15.0	1.0
13	960.0	240.0	7.0	1.0
14	700.0	160.0	0.0	1.0
15	800.0	90.0	12.0	1.0
16	1130.0	110.0	16.0	1.0
17	760.0	220.0	2.0	1.0
18	740.0	110.0	6.0	1.0
19	980.0	160.0	12.0	1.0
20	800.0	80.0	15.0	1.0

For $n=2$ the corresponding final partition is

$$C_1=\{5,6,7,9,10,11,12,13,16,17,18,19\}$$
$$C_2=\{1,2,3,4,8,14,15,20\}$$

and for $n=3$ it is

$$C_1=\{5,6,7,9,13,16,19\}$$
$$C_2=\{10,14,15,17,18,20\}$$
$$C_3=\{1,2,3,4,8,11,12\}.$$

The values for $E(C_j)$ and x_{jk} $(k=1,2,3)$ are given for $n=1,2,3$ in Table 2.

Table 2

n	j	$E(C_j)$	x_{j1}	x_{j2}	x_{j3}
1	1	99350.3	2.14	32.57	285.46
2	1	10013.3	1.24	33.40	417.58
	2	16850.1	3.60	34.21	84.56
3	1	4798.8	0.99	35.06	450.41
	2	344.3	0.52	9.88	621.23
	3	4959.4	3.42	38.89	82.53

We have had a reduction for the objective function for $n=2$ from $D=93225.1$ to $D=26863.4$ and for $n=3$ from $D=58752.0$ to $D=10102.5$. The number of passes through the observations was three for $n=2$ and for $n=3$. For $n=4$ we have got $D=1810.0$, but also one cluster C_4 with $|C_4|=l=3$, that is $E(C_4)=0$.

The computing time on a TR 440 computer for this example together with those for two other cases can be found in Table 3.

Table 3

m	l	n	time in seconds
20	3	1, 2, 3	4.8
20	4	1, 2, 3	12.6
64	4	1, 2, ..., 7	411.3

The partitions given above are not optimal ones. Starting in Step 1 with $i_0=3k$ $(k=1,\dots,6)$ better solutions are found. In the case of $n=2$ the best one is obtained for $k=6$ $(D=21404.6)$ and in the case of $n=3$ for $k=5$ $(D=4930.5)$.

5. Remarks

As indicated in [16] it is possible to reduce computing times (perhaps drastically) by calculating the solution vectors for $C_j \cup \{i\}$ and $C_j-\{i\}$ from that one for C_j by updating processes. More stable but even more cumbersome updating methods are discussed in [3, 7, 8, 11, 12, 17].

If one would choose in (3) the L_1 norm (sum of absolute deviations) or the L_∞ norm (maximum absolute deviation) instead of the squared L_2 norm (sum of squared deviations) then solutions are normally obtained through linear programming techniques [1, 2, 4, 13]. In this case, too, updating techniques are available [5] but would have to be worked out for this special purpose.

Now in the author's opinion and corresponding to his experience with real life problems from economics values of p with $p \in (1,2)$ — say $p \approx 1.3$ — are more adequate than $p=1$ or $p=2$. Choosing the L_p norm ($1<p<\infty$, $p \neq 2$) and corresponding numerical methods [9, 10, 13, 18], similar updatings seem not to be possible. Thus in this case the structure of CWDLR can be taken over and HFTI has to be replaced by a subroutine like LP from [13].

References

[1] Barrodale, I., Young, A.: Algorithms for best L_1 and L_∞ linear approximation on a discrete set. Numer. Math. *8*, 295—306 (1966).

[2] Barrodale, I., Young, A.: An improved algorithm for discrete L_1 linear approximation. SIAM J. Numer. An. *10*, 839—848 (1973).

[3] Businger, P. A.: Updating a singular value decomposition. BIT *10*, 376—397 (1970).

[4] Cromme, L.: Zur praktischen Behandlung linearer diskreter Approximationsprobleme in der Maximumsnorm. Computing *21*, 37—52 (1978).

[5] Dinkelbach, W.: Sensitivitätsanalysen und parametrische Programmierung. Berlin-Heidelberg-New York: Springer 1969.

[6] Fakiner, H., Krieger, E., Rohmeier, H.: Regional differenzierte Analyse und Prognose des Wasserbedarfs der privaten Haushalte in der Bundesrepublik Deutschland. In: Fallstudien Cluster-Analyse (Späth, H., Hrsg.). München: R. Oldenbourg 1977.

[7] Daniel, J. W., Gragg, W. B., Kaufman, L., Stewart, G. W.: Reorthogonalization and stable algorithms for updating the Gram-Schmidt QR factorization. Math. of Comput. *30*. 772—795 (1976).

[8] Lawson, C. L., Hanson, R. J.: Solving least squares problems. Englewood Cliffs: Prentice-Hall 1974.

[9] Merle, G., Späth, H.: Computational experiences with discrete L_p-approximation. Computing *12*, 315—321 (1974).

[10] Rey, W.: On least p-th power methods in multiple regressions and location estimations. BIT *15*, 174—185 (1975).

[11] Schittkowski, K., Stoer, J.: A factorization method for constrained least squares problems with data changes, part 1: theory. Preprint No. 20, Institut für Angewandte Mathematik und Statistik, Universität Würzburg, 1976. [See also: Numer. Math. *31*, 431—463 (1979).]

[12] Schittkowski, K., Zimmermann, P.: A factorization method for constrained least squares problems with data changes, part 2: numerical tests, comparisons and ALGOL-codes. Preprint No. 30, Institut für Angewandte Mathematik und Statistik, Universität Würzburg, 1977.

[13] Späth, H.: Algorithmen für multivariable Ausgleichsmodelle. München: R. Oldenbourg 1974.

[14] Späth, H.: Cluster-Analyse-Algorithmen, 2. Aufl. München: R. Oldenbourg 1977. (English translation: Cluster analysis algorithms. Chichester: Horwood 1979.)

[15] Späth, H.: Computational experiences with the exchange method applied to four commonly used partitioning cluster analysis criteria. Europ. J. Op. Res. *1*, 23—31 (1977).

[16] Späth, H.: Klassenweise diskrete Approximation. In: Numerische Methoden bei graphentheoretischen und kombinatorischen Problemen, Band 2, ISNM Vol. 46. (Collatz, L., Meinardus, G., Wetterling, W., Hrsg.). Basel-Stuttgart: Birkhäuser 1979.

[17] Stoer, J.: Einführung in die Numerische Mathematik I, 2. Aufl. Berlin-Heidelberg-New York: Springer 1976.

[18] Watson, G. A.: On two methods for discrete L_p approximation. Computing *18*, 263—266 (1977).

Prof. Dr. H. Späth
Fachbereich IV
Universität Oldenburg
Postfach 2503
D-2900 Oldenburg
Federal Republic of Germany

Eigentümer, Herausgeber und Verleger: Springer-Verlag. A-1011 Wien, Mölkerbastei 5. — Für den Textteil verantwortlich: Dr. Wilhelm Schwabl, A-1011 Wien, Mölkerbastei 5. — Für den Anzeigenteil verantwortlich: Mag. Bruno Schweder, A-1011 Wien, Mölkerbastei 5. — Satz: Austro-Filmsatz Richard Gerin, A-1020 Wien, Fischergasse 1. Druck: Paul Gerin, A-1021 Wien, Zirkusgasse 13.
Printed in Austria